EE194/290S: ML for Sequential Decision under Uncertainty **Fall 2018**

Lecture 16: Introductory Bandits 2

Lecturer: Anant Sahai/Vidya Muthukumar

Scribes: Saleh Albeaik, Jason Ramirez, Daniel Ho, Bernie Wang

16.1 Outline

- K-armed Bernoulli "Bandits"
- Exploration vs Exploitation
- Upper Confidence Bounds (UCB)

16.2 Introduction

In the multi-armed bandit (bandits) problem, several possible actions (several arms to pull) are offered sequentially (a repeated game). The player can take one action at a time and can see the reward for the action taken. The reward that would have been collected had a different action taken are not seen by the player. This lecture presents algorithms and bounds for bandits where the reward is taken from a stochastic (random, not adversarial) distribution. Namely, recall:

- **Last time:** We made sequential predictions but only received loss (reward) information for the chosen action.
- **Today:** Consider losses generated by a stochastic distribution.

16.3 Summary of take-ways from this lecture

1 Regret metric is a general metric that is suitable for all environments. This means it can potentially make us regret not having collected reward that is not achievable in the first place. Hence, regret metric should be refined in the presence of prior information (such as stochasticity as opposed to adversariality).

2 When only partial information is available exploration (learning about the environment) and exploitation (exploiting knowledge to reap reward) must essentially be balanced to achieve optimality.

16.4 K-armed Bernoulli "Bandits"

Consider the bandits reward table bellow, Table 16.1, in a limited information feedback setting:

- At round t: the player pulls arm I_t, and gets reward $x_{I_t,t}$
- The goal: to maximize expected reward $E\left(\sum_{t=1}^{T} x_{I_t,t}\right) = \sum_{t=1}^{T} \mu_{I_t,t}$

Table 16.1: Reward table for $x_{i,t} \in \{0,1\}, iid\sim Ber(\mu_i)$.

arm\round	1	2	$\cdots$	T	distribution
1	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,T}$	$iid\sim Ber(\mu_1)$
2	$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,T}$	$iid\sim Ber(\mu_2)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
k	$x_{k,1}$	$x_{k,2}$	$\cdots$	$x_{k,T}$	$iid\sim Ber(\mu_k)$

16.4.1 Learning-based benchmark

Consider the simplest scenario: suppose we know $(\mu_1, \mu_2, \cdots, \mu_k)$. In this case, the optimal strategy is to pick arm i^* with maximum mean, i.e.

$$i^* = \underset{i\in[k]}{\operatorname{argmax}}\, \mu_i$$

In this case, the corresponding mean and total expected reward for this benchmark are μ^* and $T\mu^*$, respectively.

16.4.2 "Pseudo"-regret

In this lecture, stochastic (random) environment is assumed. We therefore know that the environment belongs to the set of "nice" environments, i.e., not adversarial and not trying to get us. We hence take the liberty to refine regret into pseudo-regret. Recall that regret is a benchmark that overfits (we regret un-achievable performance) specific realization of rewards in hindsight (after the game is finished). The refinement in this section, the pseudo-regret, accounts for the fact that in a random environment, the past and the future are independent (hindsight offers no additional information). Hence, pseudo-regret, utilizes "useful" information about the randomness quality and saves us from regreting un-achievable performance (more accurate metric for regret in this case). We define the pseudo-regret as:

$$\overline{R} = T\mu^* - \sum_{t=1}^{T} \mu_{I_t}$$

16.5 Exploration vs exploitation trade-off and potential strategies

16.5.1 Exploration vs Exploitation

Now, consider the case where $(\mu_1, \mu_2, \cdots, \mu_k)$ are unknown. **For example:** k = 2 arms, $\mu_1 = 3/4$, $\mu_2 = 1/2$ (obviously, $\mu^* = 3/4$, recall distribution is iid Bernoulli). The following subsections offer a set of potential strategies to play this bandits problem with unknown means. Namely, we illustrate the trade-off between *exploration* and *exploitation* by trying to play the extreme strategies (pure exploration, and pure exploitation). We show why they fail, hence why exploration and exploitation must be balance, and in the following section, we offer remedies.

Pure exploration strategy

In a pure exploration strategy, we surrender to the fact that the means (μ_1, μ_2) are unknown, and we do not attempt to estimate those unknown quantities.

$$I_t = \begin{cases} 1 & \text{with probability } 1/2 \\ 2 & \text{with probability } 1/2 \end{cases}$$

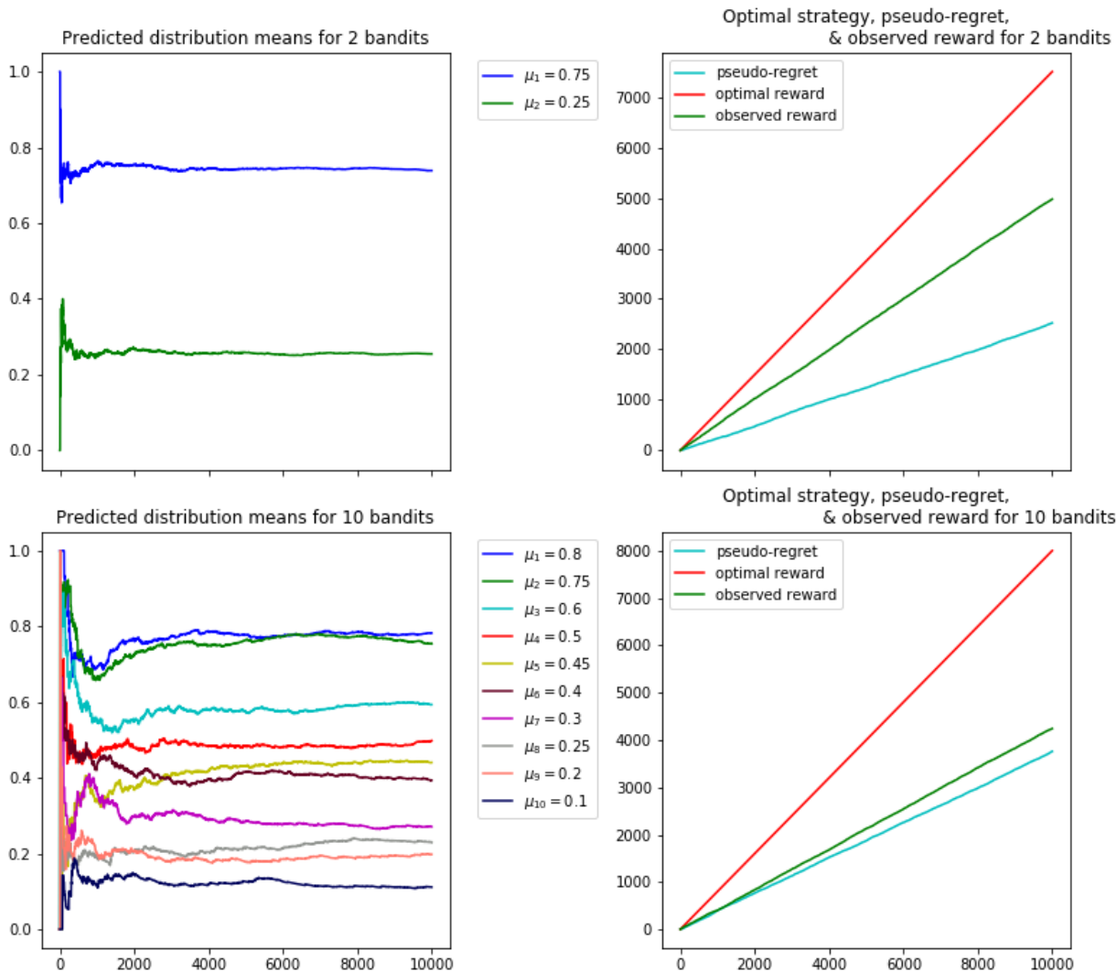

Figure 16.1: Pure Exploration Strategy for 2 & 10 bandits

With this strategy, the lower bound on pseudo-regret grows linearly with T, i.e., $\overline{R} \geq cT$ for some constant c. This can be seen in Figure 16.1

We can do better! This case is not an interesting strategy.

Pure exploitation strategy

In this strategy, we do the opposite to what we did in pure exploration. We rush through exploration (explore barely enough to observe minimal data for a cold start), and we trust our estimates of (μ_1, μ_2) too soon. That is, we use a single observation from each arm as a sample mean. We use these estimates to pick the arm with the highest sample mean.

That is, in a cold start setting (no prior information), $\hat{\mu}_1(t)$ and $\hat{\mu}_2(t)$ are unknown at $t = 0$. For a pure exploitation strategy, we establish the convention whereby we perform minimal exploration, and hence collect minimum samples to evaluate the initial sample mean estimates. That is, pull arm 1 in round 1, pull arm 2 in round 2 (and so on for the case of more than 2 arms, pull until all k arms are pulled once). If it is a tie, we explore again until we get a single sample estimate where one arm appears better than the other. We start exploitation in round 3, and hence, we have at $t = 3$, $\hat{\mu}_1(t = 3) = x_{1,1}$ and $\hat{\mu}_2(t = 3) = x_{2,2}$. For $t \geq 3$, Exploit: $I_t = argmax\{\hat{\mu}_1(t), \hat{\mu}_2(t)\}$.

This could fail! How? Consider the case where during our greedy exploration, we get $x_{1,1} = 0$ and $x_{2,2} = 1$. In this case, we keep choosing $\hat{\mu}_2(t)$ arm and keep updating its information. However, $\hat{\mu}_2(t) > 0$ for all t. Hence, the optimal arm $\hat{\mu}_1(t)$ never gets a chance after its first and only round (refer to specific example given at the beginning of section). Some results obtained by pure exploitation strategy are shown in Figure 16.2. The top row of plots demonstrates instances in which the better bandit is chosen indefinitely, the bottom left plot shows a more competitive instance, and the bottom right plot is an example of a

case where the worse bandit is chosen indefinitely.

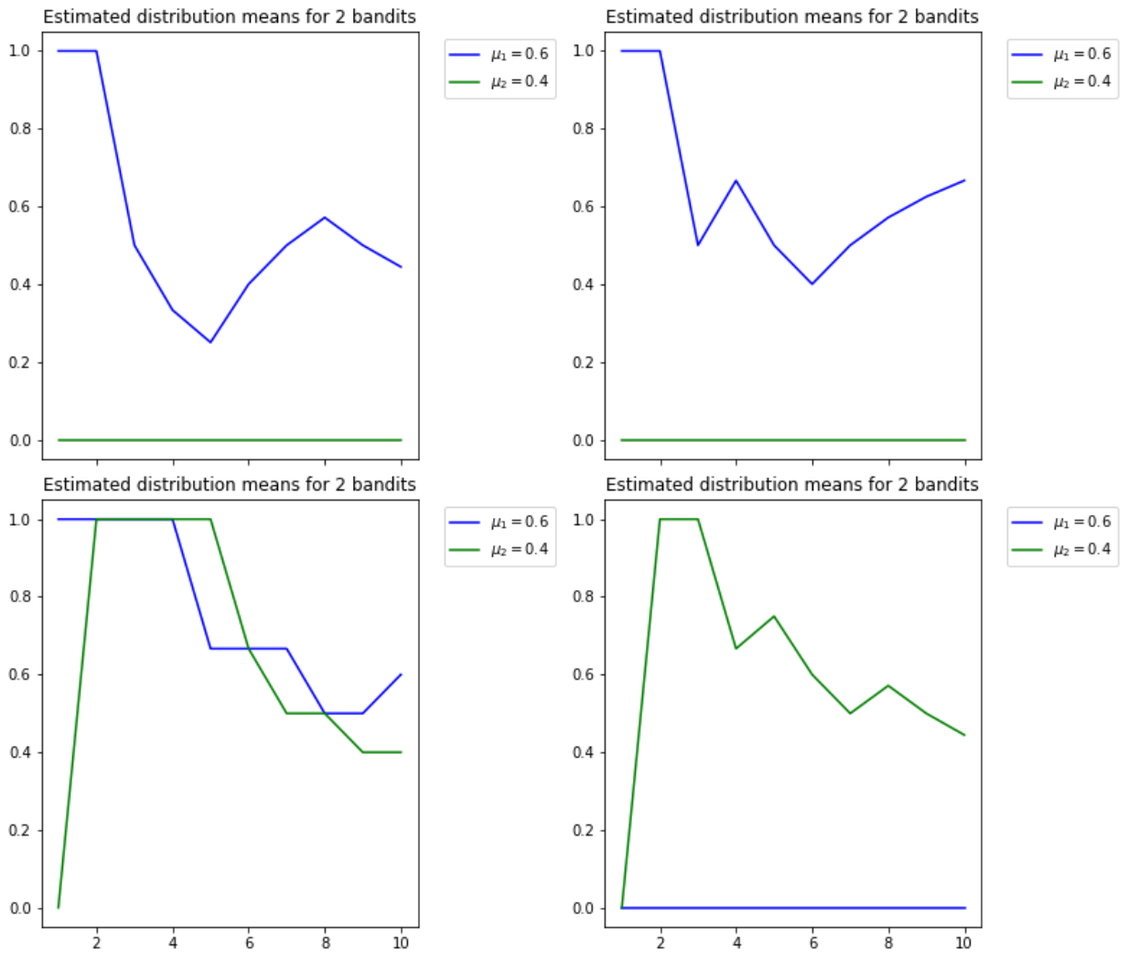

Figure 16.2: Pure Exploitation Strategy for 2 bandits

We can do better! The next plausible strategy is to combine exploration and exploitation.

16.5.2 Balancing exploration and exploitation

In the earlier section, we saw failure examples for imbalanced strategies; namely, pure exploration strategy, and pure exploitation strategy. In this section, we adjust the strategy with the realization that we can learn about the environment from experience (we thus must deviate from a pure exploration), and the realization that learning takes "time" and "effort" (and hence must deviate from pure exploitation). In the following subsections, we attempt to answer the question of how we should combine exploration and exploitation by presenting potential strategies.

Naive method: ϵ-greedy algorithm:

The simplest strategy to balance exploration and exploitation is to "pre-allocate time" for each. In the ϵ-greedy algorithm, we "pre-allocate time" by specifying a Bernoulli probability ϵ that helps us decide to explore or to exploit at each step. Consider the following algorithm:

$$I_t = \begin{cases} \underset{i \in [k]}{\text{argmin}} \{\hat{\mu}_i(t)\} & \text{w.p.} \quad (1 - \epsilon_t) \\ uniform[k] & \text{w.p.} \quad \epsilon_t \end{cases}$$

where $\epsilon_t \in [0, 1]$ for all t. We update our sample mean estimates during exploration phase. Note that $\epsilon_t = 0$ and $\epsilon_t = 1$ for all t correspond to pure exploitation and pure exploration, respectively. This strategy will keep exploring even after we have observed enough rounds;

i.e., even after our sample mean estimates, $\hat{\mu}_1(t)$ and $\hat{\mu}_2(t)$, have "almost surely converged" to the true means, μ_1 and μ_2. It also doesn't take into account the information obtained from exploiting.

Notice that ϵ depends on t. Because of the randomness of the ϵ-greedy algorithm, we want to decrease ϵ over time to reduce the chances of inefficient exploration. Consider the following extreme case, which shows why a fixed ϵ would be a bad idea. Suppose there are 50 arms and the Bernoulli distributions are parameterized as $\mu_1 = 1$ and $\mu_i = 0$ for $i \in \{2, \cdots, 50\}$. After exploring for a sufficient amount of rounds, reducing ϵ over time corresponds to being more confident about our estimates and weighting exploitation more heavily. However, if ϵ is fixed, then we would continue to explore the 49 other bad arms over time even though we may be confident about arm 1 being the best arm. Then we would continue to unnecessarily incur pseudo regret over time.

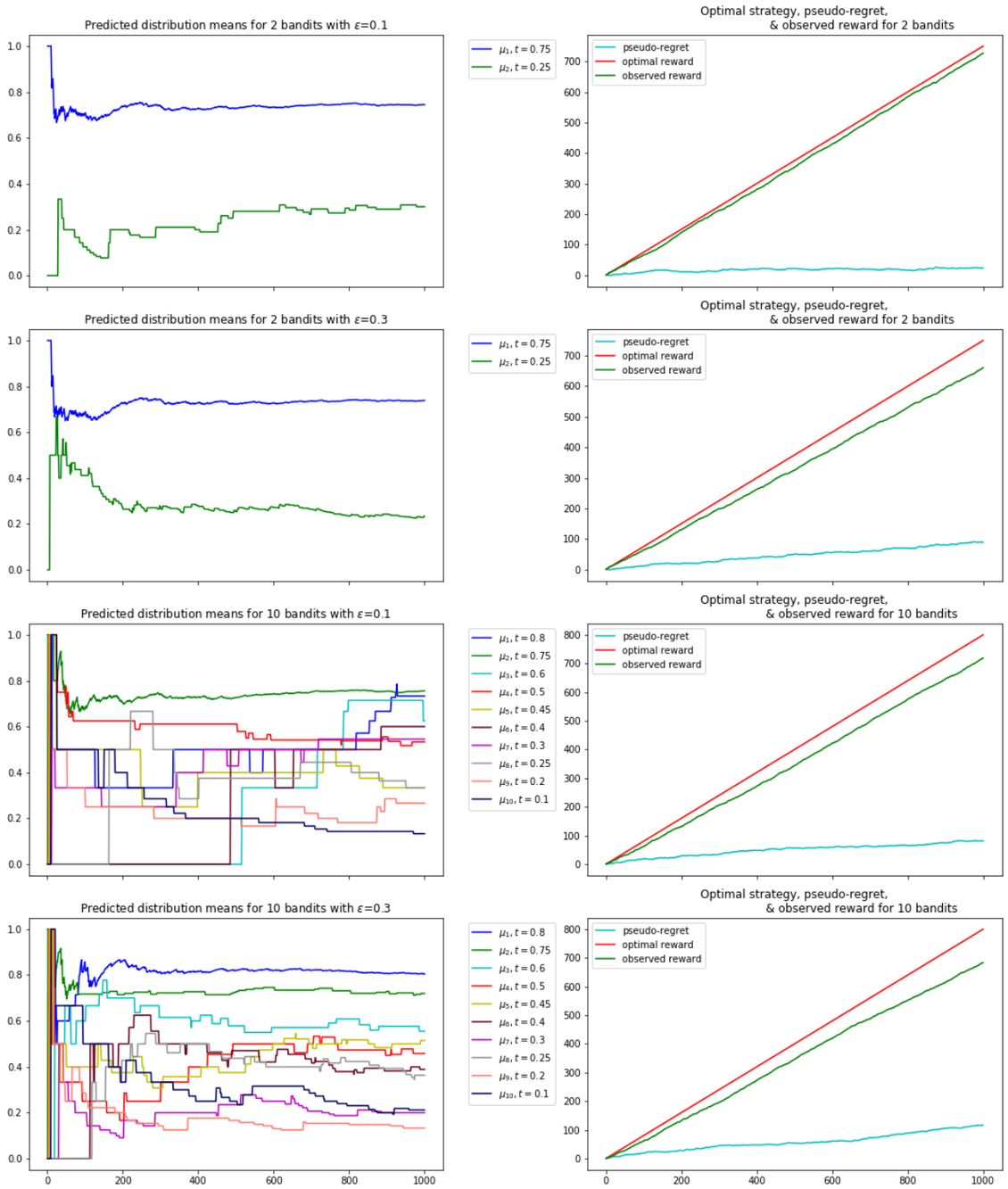

Figure 16.3: Pure Exploration Strategy for 2 & 10 bandits

A better approach: adaptive sampling and confidence bounds

Recall the trade-off between exploration and exploitation. Fundamentally, exploitation dictates what reward we get, while exploration dictates what additional information we learn.

We should ask the question: when should we stop exploring? This question is essentially equivalent to: at what point do additional information costs us more than the value of information we already have? How can we quantify these cost and value, and how can we identify this brake point in a systemic way? a reasonable answer: when our sample mean estimates, $\hat{\mu}_1(t)$ and $\hat{\mu}_2(t)$, have "almost surely converged" to the true means, μ_1 and μ_2. In this case, more exploration samples improves the sample mean in a way that does not contribute to the expected reward from exploitation. We will introduce this notion into more details, but will use it in a more relaxed way; i.e., instead of exploring until we are confident our estimates are accurate, we explore until one arm is obviously better than all other arms.

Thus we ask the question: How can we quantify "almost surely converged" and how can we embed it into our algorithm? Statisticians often quantify "almost surely converged" using *(*confidence intervals). Confidence interval is an interval around an estimated value (sample mean for example) in which the true value (the true mean we are trying to estimate) lies with sufficiently high probability. The interval tends to shrink with more information (more samples for example); i.e., we become more confident about our estimates.

We define this formally as follows:

Let $\overline{R}_T = T\mu^* - \sum_{t=1}^{T} E(\mu_{I_t,t}) = \sum_{t=1}^{T}(\mu^* - E(\mu_{I_t,t}))$, and define $\Delta_i = \mu^* - \mu_i$, the gap in max reward of arm i. I.e., how much I pay for choosing a suboptimal arm. Then, $\mu^* - E(x_{I_t,t}) = E(\Delta_{I_t})$. Hence, $\overline{R}_T = \sum_{i=1}^{k}(\Delta_i E(T_i(T)))$, where $T_i(t = T)$ is number of time I pulled arm i by time t. We want to minimize the number of times we pull the sub-optimal arm, while making sure we explore enough to find the true optimal arm. Consider the second diagram in Figure 16.3, where the confidence interval of the third bandit is much larger than that of the first and second bandit. Intuitively, because the confidence interval for the third bandit is much larger, we are more unsure about its maximum expected reward compared to the other two arms. Consequently, we should incentivize exploring arm 3 to obtain more information to exploit in future rounds. Next, we try to do this in a principled way.

We need to answer the following questions and and design an algorithm with the follwoing properties:

Question: at round t, we want to know $\hat{\mu}_i$ for arm i, and $T_i(t-1)$ the number of times we had pulled arm i by now.

Question: how good is our estimate so far (at time t)? to what extent can we say $\hat{\mu}_i \approx \mu_i$?

Goal: optimizing collected reward (exploitation)

Goal: optimizing our estimate quality of $\hat{\mu}_i$ (exploration)

Decision: the algorithm must tell us the optimal sequence of arms to pull

Consider the following algorithm:

$$I_t = \underset{i \in [k]}{\operatorname{argmax}} \{\hat{\mu}_i(t) + C.I._i(\delta)\}$$

where $C.I._i(\delta)$ is the δ-confidence-interval for our estimate of $\hat{\mu}_i(t)$. This algorithm basically incentivizes exploration until our estimates have "almost surely converged".

Definition, Confidence Interval: $C.I._i(\delta) := \epsilon$ for which $Prob(\hat{\mu}_i(t) - \mu_i > \epsilon) \leq \delta$

For Bernoulli setting: $\hat{\mu}_i(t)$ = mean of $T_i(t-1)$ $Ber(\mu_i)$ random variables

$\rightarrow Prob(\hat{\mu}_i(t) - \mu_i > \epsilon) \leq e^{\frac{-\epsilon^2 T_i(t-1)}{8}}$ (from Hoeffding's Inequality)

Exercise: let $(\delta_t = t^{-\alpha})$, then $C.I._i(\delta_t, t) = \sqrt{\frac{8(\alpha ln(t))}{T_i(t-1)}}$

How do we choose α?

Why should δ_t decrease with time? Decreasing δ_t over time is related to the idea of simulated annealing in which the temperature parameter follows an annealing schedule and basically cools down over time, indicating the probability of accepting worse solution decreases over time. In this case, suppose δ_t was independent of time. Then in the long

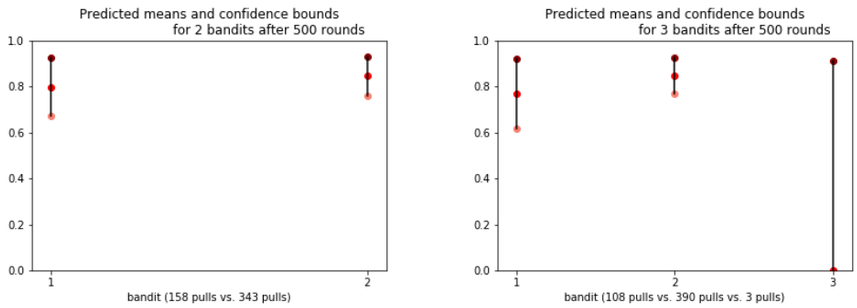

Figure 16.4: Confidence bounds for 2 & 3 bandits after 500 rounds

run, it's possible to converge to sub-optimal estimates for all μ_i. For example, consider the extreme case where $\delta_t = 1$, then $\mathbb{P}[\widehat{\mu}_i(t) - \mu_i > \epsilon] \leq \delta_t = 1$ would work for any ϵ and the algorithm would reduce to pure exploitation. In figure 16.4 where the arms are parameterized by mean 0.3, 0.5, and 0.51 respectively, we see that for large fixed delta values, the algorithm can converge to the wrong predicted means. Thus, it is important to make δ_t decrease with time so that the algorithm can effectively explore arms whose estimates have higher degree of uncertainty.

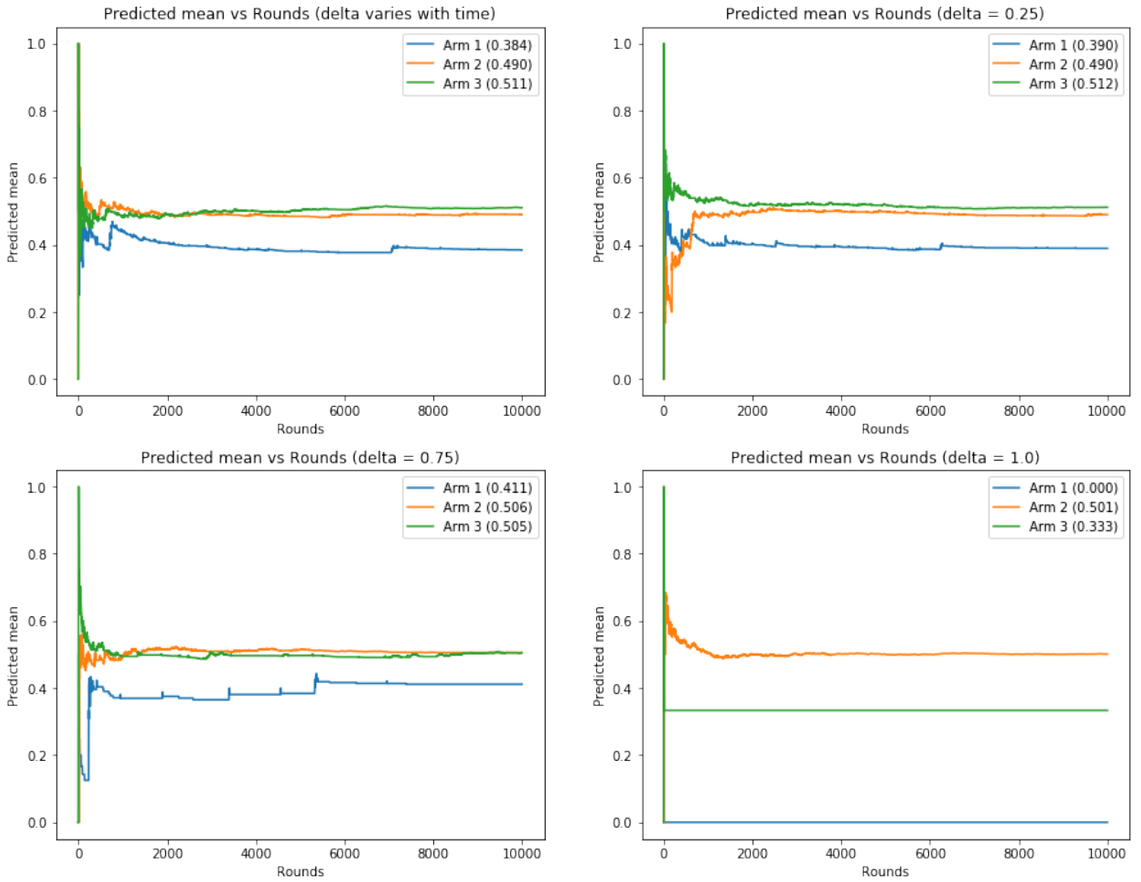

Figure 16.5: Comparison of fixed vs time-varying delta

16.6 Upper confidence bound (UCB) algorithm:

In this section, we wrap up what we have learned so far into one elegant algorithm. Consider the following algorithm:

$$I_t = \underset{i \in [k]}{\operatorname{argmax}} \left\{ \hat{\mu}_i(t) + \sqrt{\frac{8(\alpha ln(t))}{T_i(t-1)}} \right\}$$

In this algorithm, we play arm i until we are sufficiently certain it is not the right arm to play. The UCB Strategy can be seen in Figure 16.6.

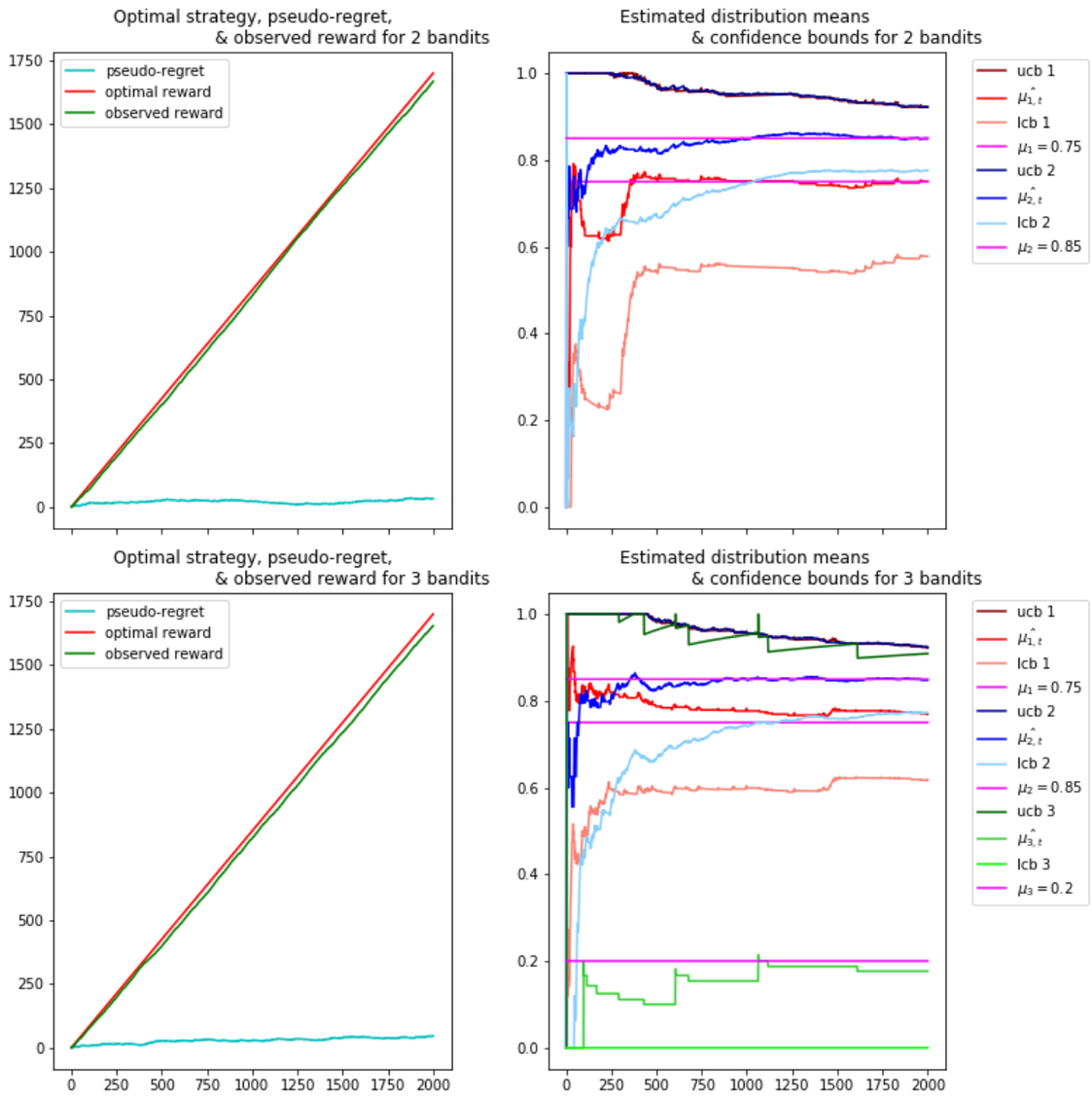

Figure 16.6: UCB Strategy for 2 & 10 bandits ($\alpha = 0.15$)

Theorem Pseudo-regret bound (holds for $\alpha > 2$).

$$\overline{R} \leq \sum_{i:\Delta_i>0} \left[2\frac{8\alpha ln(T)}{\Delta_i} \right] + constant$$

Notice that the upper bound on regret is larger for smaller Δ_i. This is because we need to explore longer to be able to start distinguishing the more similar arms. That is, we need to increase T that appears on the numerator. Otherwise, we risk settling on the wrong arm too early, and hence incur large regrets over all.

How many times do we eventually expect to pull arm i?

$\rightarrow$ for arm i, $E(T_i(T)) \leq 2\frac{8\alpha ln(T)}{\Delta_i^2} + constant$

let $u_i = 2\frac{8\alpha ln(T)}{\Delta_i^2}$

What if we started to sample arm i more than expected? That is, we sampled arm $i > u_i$ times.

UCB would have select arm i over arm i^* only if:

a) $\hat{\mu}_i^* + \sqrt{\frac{8\alpha ln(T)}{T_i^*(t-1)}} < \mu^*$

b) $\hat{\mu}_i - \sqrt{\frac{8\alpha ln(T)}{T_i^*(t-1)}} < \mu_i$

In those cases, the small probability of confidence interval failure (the true mean was actually outside the confidence interval) had happened.

Define events A(a) and A(b) as follows:

- $A(a) \rightarrow T_i(t-1) \geq u_i$
- $A(b) \rightarrow T_i^*(t-1) \geq u_i^*$

Then, we can say: $P(A(a)) + P(A(b)) \leq t^{-\alpha}$

In this case, we just trust our estimates and our bounds stay intact.